

Team 3 "iNODES for Wi-Fi Network Enhancement"

Luke D Heide 11/14/2011

Team Sponsors: ArcelorMittal and RCPD



Application Note

ANDROID OPERATING SYSTEM APPLICATION DEVELOPMENT WITH MYSQL SERVER CONNECTIVITY

Abstract: iDOCENT is a phone application created to assist navigation throughout buildings on Michigan State University's campus. The software application is designed for any user unfamiliar with Michigan State University's campus and also includes a universal design for the usability of handicapped individuals. This application note will describe the process of setting up a computer to begin application development for the android system, and properly setting up a MYSQL server for iDOCENT to download information from.

Table of Contents

Introduction.....	3
Key Terms	3
Getting Started: Android iDOCENT Application Development.....	4
Required Tools For Development Computer.....	4
Step 1: Download the Java SDK	4
Step 2: Download the Eclipse IDE	4
Step 3: Download the Android SDK Starter Package	4
Step 4: Installing the ADT Plug-in for Eclipse	5
Step 5: Adding Platforms and other Components	5
Step 6: Importing and Running the iDOCENT Code	5
Functional iDOCENT Application: Setting up Server and Server Code	6
Step 1: Download and Install MYSQL and Utilize the Server Dump File	6
Step 2: Importing the Path Server Code to Eclipse.....	7
Step 3: Changing the Client and Server code for your specific Server	7
InetRLookup.java	7
NavigationDownloader.java	8
RoomDownloader.java	8
DBManager.java.....	9
Step 4: Loading the iDOCENT application to your Android Phone and Testing	9
Trouble Shooting and Further Resources	10
Configuring Your Router for Outside Applications	10
Further Resources for Understanding the Code	11
Further Resources for Android Development Tutorials.....	11
Conclusion	11
Bibliography.....	11

Introduction

The iDOCENT smart phone application was a very important aspect of the ECE 480 Team 3 project. The Android operating system and smart phones in general are continually updating and adding improved features that could benefit the iDOCENT application in the future. In order to allow future ECE 480 teams to easily transition into the further development of the iDOCENT smart phone application, this application note will highlight the important steps needed to be taken to set up the development environment and MYSQL server. The previous iDOCENT team has important and helpful application notes to help in further understanding the code of the iDOCENT application and Server application, but lacks an application note highlighting common technical problems for getting started with the applications. Once the iDOCENT application and Server application are completely functional and ready for debugging, a better understanding of the code is necessary and useful, but getting to this point could prove to be frustrating. For our team this year, getting the iDOCENT application and Server application functional involved a greater amount of time than desired and back and forth contacts to the previous team. This application note will serve the purpose of bridging the gap between the previous semester application notes and this semester's application notes, and will decrease the amount of time future teams will spend getting the iDOCENT code and Application code functional so debugging and understanding can occur quicker. This development tutorial is specific to the iDOCENT application and server, and will be far more helpful and application specific than general Android Development tutorials.

Key Terms

AVD - Android Virtual Device: Development tool used for debugging on a "virtual" Android device. Used for debugging before putting on an actual Android Hardware Device.

IDE – Integrated Development Environment: a program designed to assist a programmer to make programming and executing applications easier on a specific platform or platforms.

SDK – Software Development Kit: a set of tools that assist in the development of an application.

Wi-Fi – Wireless Fidelity: The protocol most frequently used for wide local area networks.

Getting Started: Android iDOCENT Application Development

Required Tools for Development Computer

****Note**** All download links mentioned in a section can be found at the end of the section. The chosen development computer will need to be kept away from the Engineering Building Wi-Fi network, must be able to connect to a different network, and will need to run the server application code that will be described in the next section. Although these requirements do not apply to the actual application code, it is nice to have the application code debugging available on the same computer as the server code.

Step 1: Download the Java SDK

The SDK needed for Android development is the Java SDK, which is named JDK, or Java Development Kit. The JDK is the set of software development tools that allows for the creation of all Java applications, including Android smart phone applications. As of writing this application note, the latest version of the JDK is version JDK7 update 1 (JDK-7u1). You can download the latest version of the JDK for free at the oracle website. It is important to download the correct version for your operating system (windows 7 32 bit, windows 7 64 bit, MAC OS, Linux act.)

JDK Download: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Step 2: Download Eclipse IDE

The recommended development environment for programmers new to Android development is the Eclipse IDE. This is because Android Development Tools (ADT) can easily be installed on eclipse via a plug-in which will be discussed in later steps. Eclipse IDE can be downloaded for free at the Eclipse website.

Eclipse IDE Download: <http://www.eclipse.org/downloads/>

Step 3: Download the Android SDK Starter Package

The SDK Starter Package for Android does not include the full development environment, but only the core SDK tools. This is enough to get the iDOCENT phone application running and debugging on an Android device. The latest SDK starter package can be downloaded for free at the Android Developer website. It is important to make note of where you download the SDK starter package directory, as it will be needed when setting up the Eclipse IDE for the first time.

SDK Starter Package: <http://developer.android.com/sdk/installing.html>

Step 4: Installing the ADT Plug-in for Eclipse

The ADT plug-in, as mentioned above, works with the Eclipse IDE and JDK to create an environment which enables the programmer to build Android smart phone applications. It allows the programmer to quickly set up new Android projects, create application user interfaces, and debug the projects using the Android SDK tools. To download the ADT plug-in, you first must enter the Eclipse IDE and select "Help" -> "Install New Software....". Click "Add" in the top right corner. In the "Add Repository" dialog, enter "ADT Plug-in" for the name field, and enter the ADT URL for the Location (found at end of section). Click "OK". It is possible that using https in the URL instead of http will cause a problem. If the problem insists, use http.

To configure the new ADT Plug-in, select "Window" -> "Preferences" and select "Android" from the left panel. For the SDK location in the main panel, click Browse and locate the SDK location referenced in Step 3. Hit apply and then OK.

ADT URL: <https://dl-ssl.google.com/android/eclipse/>

Troubleshooting Information: <http://developer.android.com/sdk/eclipse-adt.html#installing>

Step 5: Adding Platforms and Other Components

The last step in setting up the development computer is installing the correct Android SDK and installing an AVD. The iDOCENT application runs on the Android platform 2.1 update 1. This is not the newest version of Android, but all important code features, including the Wi-Fi management code stay the same in the new versions, so this does not pose an issue when transferring to an Android phone. To install the Android SDK 2.1 platform, select "Window" -> "Android SDK and AVD Manager". Use the graphical interface to select Android 2.1 and click install. For more information on each specific Android platform, including the latest Android platform releases, you can visit the develop site page, but the Android 2.1 platform is what the iDOCENT application requires.

To install an AVD to debug the iDOCENT code, in eclipse select "Window"-> "Android SDK and AVD Manager" -> "Virtual Devices" -> "New....". Type any name for the AVD and choose the target. Since iDOCENT runs on Android SDK 2.1, select SDK 2.1. Select Create AVD. You may want to choose the version of Android the test phone is using instead of 2.1.

Android Platform Information: <http://developer.android.com/sdk/installing.html>

Step 6: Importing and running the iDOCENT Code

The first step in getting the iDOCENT code running in your eclipse IDE is to download the code from the original iDOCENT team's application code from the ECE 480 website, or obtain the latest code from Fall 2011 team's website. When the .zip file is downloaded, extract it and remember the location of the iDOCENT directory. In Eclipse select "File" -> "Import". Under the "General" folder there will be an "Existing Projects into Workspace". Select this option and hit "Next". Browse to the iDOCENT directory just downloaded and hit OK. If all steps were completely properly, the iDOCENT application should build and compile without any errors. If

there are errors, it is most likely due to a mistake in SDK downloads and installations from previous steps.

To run the code in the AVD, right click the iDOCENT folder in the project explorer on the left side of the eclipse IDE and select "Run as" -> "Android Application". The AVD will boot up (this may take up to several minutes) and the iDOCENT application will start as soon as you unlock the virtual phone. The iDOCENT app should be running, and you will be able to maneuver through the user interface. If you select a room to navigate to, the AVD will crash, as Wi-Fi capabilities are not supported with the AVD's, and the Java Server is not setup yet. You are now ready to begin setting up the MYSQL server and Java application code.

Source Code Download:

<http://www.egr.msu.edu/classes/ece480/capstone/spring11/group02/documents/default.html>

Functional iDOCENT Application: Setting up Server and Server Code

The iDOCENT application that was set up in the "Getting Started" section requires server connectivity to download the necessary information about the access point location and transmitting properties in the Engineering Building. At the end of this section you will have a functional iDOCENT application for the Engineering Building at Michigan State University. It is important that this server is not run from the Engineering Building Wi-Fi. This server will need to be running away from the network.

Step 1: Download and Install MYSQL and Utilize the Server Dump File

The MYSQL installation is fairly straight forward, and there is not any extra configuration you must add like the eclipse installation. "MYSQL is a relational database management system that runs as a server providing multi-user access to a number of databases" (MYSQL Wiki). You can get a free copy of the MYSQL program at the MYSQL sight. Go through the automatic installation guide (make sure you are installing under administrator account if on Windows), remember the username and password you create for the server (this will be in the code) and remember the port used in configuration (the default port is 3306).

Once MYSQL installed, you will create a new "schema" and name it ebfloor1. You will then navigate to the Spring 2011 or Fall 2011 team website to download the iDOCENT Server source code. As before with the iDOCENT application source code, extract it to a location you will remember. Back in MYSQL, you select the newly created schema, and open and execute the dump file (.dmp) located in the Path Server directory just downloaded. The database containing room and access point information is not available on your server.

Source Code Download:

<http://www.egr.msu.edu/classes/ece480/capstone/spring11/group02/documents/default.html>

Step 2: Importing the Path Server code to Eclipse

In step 1, the Path Server directory was downloaded, which contains the server application code. This code must be imported to run in the Eclipse IDE. In eclipse go through the automatic setup of creating a new Java Project by selecting "File" -> "New" -> "Java Project". The name of the project can be "Server". When the project creation wizard is complete, right click on it in the Package Explorer pane on the right side of eclipse, and select "Import". Under general pick "file system". Find the directory to Path Server and select it. Click the checkbox to the left to import all the files, and then click "Finish". The project will show that there are compile errors; this is because you need to configure the build path for the MYSQL connector. To do this, right click on the project and select "Build Path" -> "Configure Build Path". Under "Libraries" select "Add External Jars". Search for mysql-connector-java-5.1.15-bin.jar in Code\PathServer\jdonofrio@72.2.44.121.49\lib and select it. Hit ok. This will clear up all compile issues, and the server code is ready to run.

Step 3: Changing the Client and Server code for your specific Server

There are a total of four .java files that will need to be updated to make the code specific to your server. Within the client code there are: InetRLookup.java, NavigationDownloader.java and RoomDownloader.java. Within the server code there is just DBManager.java. To access these java files, look to the Package Explorer section of Eclipse on the left side of the IDE. Under each Project folder there is a src folder. Under the src folder in the iDOCENT application there is a package com.iDOCENT. Expand this folder for the client .java files. Under the src folder in the server project there is a patherserver package. Expand this package for the .java files for the server side.

To edit the .java files, simply double click the specific .java file in the package explorer. Below are the sections of code that will need to be changed for each .java file.

InetRLookup.java

```
/**Internet Router Lookup to download AP location*/
public class InetRLookup {
    private String mac;
    //private final String DNS = "192.168.1.68";
    private final String DNS = "99.82.240.243";
    //private final String DNS = "480team2.dyndns-server.com";
    private final int Port = 1024;
```

the line:

```
private final String DNS = "99.82.240.243";
```

This line will need to be changed to reflect the IP address of the computer running the server. Note that this is a public IP address and not your local IP. To obtain this IP address you can go to a website like whatismyipaddress.com

NavigationDownloader.java

```
/**download the route to the destination*/  
public class NavigationDownloader {  
    //private final String DNS = "192.168.1.68";  
    private final String DNS = "99.82.240.243";  
    private final int Port = 1024;
```

the line:

```
private final String DNS = "99.82.240.243";
```

Again this will need to be changed to reflect the server PC's IP address.

RoomDownloader.java

```
//private final String DNS = "192.168.1.68";  
private final String DNS = "99.82.240.243";  
private final int Port = 1024;
```

the line:

```
private final String DNS = "99.82.240.243";
```

Again this will need to be changed to reflect the server PC's public IP address.

DBManager.java

```
public void Connect(){  
    //  
    try{  
        con = DriverManager.getConnection(url, "root", "team03");  
        Connected = true;  
    }  
    catch(SQLException e){  
        System.out.println("SQLException: " + e.getMessage());  
        System.out.println("SQLState: " + e.getSQLState());  
        System.out.println("VendorError: " + e.getErrorCode());  
    }  
}
```

This line:

```
con = DriverManager.getConnection(url, "root", "team03");
```

The "root" word will be replaced with the username you selected for your server (MYSQL default is "root"). The "team03" word will be replaced with the password you selected for your server.

Step 4: Loading the iDOCENT application to your Android Phone and Testing

The final step in getting a functional iDOCENT application on your Android phone is using eclipse to load the Application to your phone. First you must setup your phone to allow applications from unknown sources. On your Android phone go to "Settings" -> "Applications" -> "Development" and enable "USB debugging". The easiest way to do this is to plug your Android phone into the computer running eclipse via a USB port and use it for debugging. Second, you must set the AndroidManifest to allow for debugging in eclipse. To do this, locate the AndroidManifest.xml in the Package Explorer and double click it. In the "Application" tab set the "Debuggable" field to "true".

Before you connect your Android phone to your computer, make sure your system will recognize your device for debugging. On Windows, you will need to make sure you have the USB driver for ADB. If you're using the Nexus One or Nexus S, you can download the Google Windows USB Driver. Otherwise, use the OEM specific USB driver from the USB driver link below. If you are developing on a MAC, you can skip this step. For more tutorials for operating systems such as Ubuntu Linux, you can visit the development page tutorial.

Now you can connect your phone to your computer running eclipse. Before you run the iDOCENT application on your phone, you will want to make sure the Server application is running. To do this, right click the server folder and select "Run As" -> "Java Application". Once

the server application is running, right click the iDOCENT project and select "Run As" -> "Android Application". If your phone has a working Wi-Fi connection, the console log at the bottom of eclipse should report a connection from the phone. If the phone cannot connect to the server, you may need to configure your router to allow connections from outside devices. Information to do this can be found in the trouble shooting section.

Once the phone successfully connects to the server, the application will work in the Engineering Building. Until you get the phone in the engineering building, it will always crash when selecting a room, as the access points it is looking for are not available. Remember to always have the Java application running when you want to use iDOCENT application.

Google Windows USB Driver: <http://developer.android.com/sdk/win-usb.html>

OEM Specific Drivers: <http://developer.android.com/sdk/oem-usb.html>

Extra help for Linux Systems: <http://developer.android.com/guide/developing/device.html>

Trouble Shooting and Further Resources

Configuring your Router for allowing Outside Applications

The IP address that was setup in the client code actually establishes a connection to the router of the computer that the server is running off of. Most all routers have built in firewalls to block any intrusions to a specific computer on the local network. For the router to understand to allow connections for applications requesting access to port 1024, you will need to set it up to forward incoming traffic requests on port 1024 to your computers local IP address. Your computers local IP address can be found going to "Run" -> typing "cmd" -> then "ipconfig" in the command prompt, or going to your router homepage. To go to your router's homepage, you open your default internet browser and enter the router's IP. Refer to the user manual of your specific router on how to log in.

From your router's login page, you will be able allow specific applications to connect to the server computer through port 1024. Every router is different, and the best way to find information for your specific router is to do a Google search.

If after you set up the router configuration and the application still cannot connect to the server, you may have to look into your personal firewalls on your computer. A good way to see if these firewalls are the problem is to temporarily disable them and try to connect when they are disabled.

Further Resources for Understanding the Code

For a further understanding of the iDOCENT application phone code and using Java code to manage the Wi-Fi capabilities of your smart phone, view Matthew Gottshal's application note from ECE 480's Spring semester 2011.

For a further understanding of the Java Network Server code, and understanding how the code allows for connections from the iDOCENT application, view Jacob D'Onofrio's application note from ECE 480's Spring semester 2011.

Matthew Gottshal's application note:

http://www.egr.msu.edu/classes/ece480/capstone/spring11/group02/documents/matt_appnote.pdf

Jacob D'Onofrio's application note:

http://www.egr.msu.edu/classes/ece480/capstone/spring11/group02/documents/jake_appnote.pdf

Further Resources for Android Development Tutorials

The best resource for Android Development tutorials would be the official Android Developers home page.

Android Developers Home Page:

<http://developer.android.com/index.html>

Conclusion

The tools and resources needed to begin iDOCENT application development are the Java SDK, Eclipse IDE with Android SDK starter kit and AVD, Android ADT plug-in for Eclipse and MYSQL program for creating the server. After reading and following the steps for this application note, the iDOCENT and Server application should be ready for testing use in Michigan State University's Engineering Building.

Bibliography and Sources

<http://developer.android.com/index.html>

Basic steps of setting up the Eclipse IDE were paraphrased from this website, but added were the iDOCENT specific setup steps.

<http://whatismyipaddress.com/>

<http://www.egr.msu.edu/classes/ece480/capstone/spring11/group02/documents/default.html>

<http://en.wikipedia.org/wiki/MySQL>

http://en.wikipedia.org/wiki/Integrated_development_environment

<http://en.wikipedia.org/wiki/JDK>

<http://en.wikipedia.org/wiki/SDK>